APPLICATION FOR UNITED STATES PATENT

in the name of

Joseph G. Barrett, Christopher J. Wright, and Victor R. Parente

of

America Online, Inc.

for

SECURING AN ACCESSIBLE COMPUTER SYSTEM

SECURING AN ACCESSIBLE COMPUTER SYSTEM

INSA1
INSA2

TECHNICAL FIELD

This invention relates to securing an accessible computer system. More particularly, this invention relates to detecting and preventing denial of service attacks on a computer system.

BACKGROUND

Accessible computer systems have proven susceptible to various attacks by computer hackers. In a type of computer attack known as a denial of service attack, a hacker attempts to deny service to legitimate users of online computer services. For instance, a hacker may send a high number of illegitimate access requests to an accessible computer system of an online computer service, causing the computer system to dedicate its resources to handling the illegitimate access requests rather than handling legitimate access requests from legitimate users. In this manner, legitimate users may be denied access to the online computer service because of the influx of illegitimate access requests sent by the hacker. This type of attack is commonly known as a synchronize (SYN) flood.

Another type of a computer attack occurs when a hacker attempts to gain unauthorized access to an online computer service. In this type of attack, the hacker uses a client to attempt to establish an unauthorized connection with the online computer service. Using a known logon identification, the hacker attempts to crack the password associated with the known logon identification by using a computer program that associates several passwords with the logon identification in rapid succession, repeatedly attempting to establish a connection to the computer service using the known logon identification and one of the associated passwords. This type of attack may tax processing resources to effectively deny legitimate users access to the online computer service.

When subject to such attacks, accessible computer systems may be forced to cease operation.

SUMMARY

In one general aspect, securing an accessible computer system includes monitoring a computer system for connection transactions between at least one access requestor and at least one access provider and denying access by the access requestor to the access provider when a number of connection transactions initiated by the access requestor exceeds a configurable threshold number during a first configurable period of time.

Embodiments may include one or more of the following features. For example, the monitoring may include detecting connection transactions initiated by the access requestor. The monitoring also may include counting the number of connection transactions initiated by the access requestor during the first configurable period of time, and comparing the number of connection transactions initiated by the access requestor during the first configurable period of time to the configurable threshold number.

The monitoring may include detecting connection transactions between at least one Internet protocol address and the access provider. The monitoring also may include counting the number of connection transactions initiated by the Internet protocol address during the first configurable period of time, and comparing the number of connection transactions initiated by the Internet protocol address during the first configurable period of time to the configurable threshold number.

Where the access requestor is a client and the access provider is a host, the monitoring may include detecting connection transactions between at least one client and at least one host. Where the access requestor is a client and the access provider is a host, the monitoring also may include detecting connection transactions between the access requestor and a group of access providers. The connection transactions may include connections made using TCP. Detecting connection transactions may include identifying the Internet protocol address through the use of a header attached to a message representing the connection transaction being detected.

The denying of access may include denying access to the access provider by the access requestor for a second configurable period of time. The denying of access also may include denying access to the access provider by the access requestor for a second configurable period of time after detecting a most recent connection transaction initiated by the access requestor. The denying of access may further include resetting the second

configurable period of time after detecting a new connection transaction initiated by the access requestor during the second configurable period of time.

The general and specific aspects may be implemented using a system or method or combination of system and method.

The details of one or more embodiments are set forth in the accompanying drawings and the description below. Other features and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

Fig. 1 is a block diagram that illustrates a physical level of an accessible computer system.

Fig. 2 is a block diagram that illustrates a logical level of the accessible computer system of Fig. 1.

Fig. 3 is a block diagram that illustrates components included in a switch, such as those shown by Figs. 1 and 2.

Fig. 4 is a block diagram that illustrates components included in a monitoring component of the switch of Fig. 3.

Fig. 5 is a block diagram that illustrates components included in a blocking component of the switch of Fig. 3.

Fig. 6 is a flowchart of a process for securing an accessible computer system, which may be performed by the systems shown by Figs. 1-5.

Fig. 7 is a flowchart of a process for monitoring the computer system for connection transactions as part of the process of Fig. 6.

Fig. 8 is a flowchart of a process for controlling access to access providers as part of the process of Fig. 7.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

Fig. 1 is a block diagram that illustrates the physical level of an accessible computer system 100. Fig. 1 shows multiple access requestors 110, the Internet 130, multiple routers 150, switch 170, multiple access providers 190, and multiple communication links 120, 140, 160, and 180.

An access requestor 110 may include a client, and may be embodied in a general-purpose computer (e.g., a personal computer), a special-purpose computer, a workstation, a server, a personal digital assistant, an electronic organizer, a mobile phone, a pager, a device, a component, or other physical or virtual equipment or some combination thereof, any of which may be programmed or configured to respond to and execute instructions in a defined manner. Access requestors 110 are connected to the Internet 130 by communication links 120.

The Internet 130 is an example of a delivery network that may be used to enable communications to/from access requestors 110. Other examples of a delivery network may include the World Wide Web, wide area networks (WANs), local area networks (LANs), analog or digital wired and wireless telephone networks (e.g. Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), and Digital Subscriber Lines (xDSL)), radio, television, cable, satellite, and/or any other delivery mechanism for carrying data. The Internet 130 is generally connected to one or more routers 150 by communication links 140.

Each router 150 generally includes a computer processor, computer software, a hardware device, other physical or virtual equipment or some combination of these elements that is capable of receiving, processing and transmitting information. In general, each router 150 routes communications between one or more access requestors 110 and one or more access providers 190. Communications received from an access provider 190 are generally routed to an access requestor 110 through the Internet 130. Communications received from an access requestor 110 are generally routed to an access provider 190 through a switch 170. More specifically, each router 150 receives a data packet and/or data request from access requestor 110 and routes the data packet and/or data request to one or more of the access providers 190 based on predefined criteria or algorithms. The routers 150 are connected to one or more switches 170 by communication links 160.

Switch 170 generally includes a hardware component and a software component. It is capable of receiving a unit of data and of transmitting the received data to one or more access providers 190 or routers 150 based on predefined criteria or algorithms. Switch 170 may perform load balancing algorithms such as hashing techniques to avoid overwhelming any particular router 150 or access provider 190. Switch 170 also may perform the functions of the router 150 as a separate or integrated component or device. Additionally or alternatively,

switch 170 may include one or more processors and one or more storage and memory devices, such as internal memory. The switch 170 is connected to multiple access providers 190 by communication links 180.

An access provider 190 may be any software or hardware capable of providing access by an access requestor 110 to desired information or services. For instance, an access provider 190 may include a host, and it may be embodied in a general-purpose computer (e.g., a personal computer) or a special-purpose computer capable of communicating with one or more access requestors 110 by responding to and executing instructions in a defined manner. Other examples of an access provider 190 include a special-purpose computer, a work station, a server, a device, a component, other physical or virtual equipment or some combination of these elements that is capable of responding to and executing instructions as described.

Communication links 120, 140, 160 and 180 may include, for example, a wired, wireless, cable or satellite communication pathway.

Fig. 2 is a block diagram that illustrates a logical level of the system 100 illustrated in Fig. 1. Fig. 2 shows multiple access requestors 110, switch 170, and multiple access providers 190. In this figure, switch 170 may be representative of one or more of Internet 130, router 150 and switch 170, or some combination there between such as that described in Fig. 1.

An access requestor 110 is generally used to establish a physical or non-physical electronic connection with an access provider 190. Connections may be established on various levels using various protocols. For instance, a connection may be established on Level III (e.g., a packet based level), on Level IV (e.g., a protocol data unit based level with flow control and error correction) or on some other level using an appropriate protocol capable of establishing a connection between an access requestor 110 and an access provider 190. More specifically, examples of protocols include Transmission Control Protocol (TCP), Internet Protocol (IP), TCP/IP, User Datagram Protocol (UDP), and UDP/IP.

Access protocols are observed to establish a connection. Under an exemplary Level IV protocol, an access requestor 110 sends an access request through switch 170. The request is routed to one of the access providers 190, which responds to the access request by sending an acknowledgement that is routed back to the access requestor 110 through switch 170. When the access requestor 110 receives the acknowledgement sent by the access

provider 190, the access requestor 110 generates an acknowledgement that is sent back to the access provider 190 through switch 170. The completion of this transaction establishes a connection between the access requestor 110 and the access provider 190.

For purposes of this detailed description, the term connection transaction is used to describe one or more of sending, receiving, or exchanging the units of data necessary to use a protocol (e.g., TCP, IP, UDP, TCP/IP, and UDP/IP) to establish a communications link (e.g., wired, wireless, cable, and satellite) between the access requestor 110 and access provider 190. One example of a connection transaction results in a TCP connection between the access requestor 110 and the access provider 190, where procedures to establish a connection transaction use the synchronize (SYN) control flag and involve an exchange of three messages. In this example, an access requestor 110 sends an access request (SYN REQ) to an access provider 190 through switch 170. The access provider 190 responds to the access requestor 110 through switch 170 with an acknowledgement (SYN ACK). Then, the access requestor 110 sends an acknowledgement (ACK) to access provider 190 via switch 170. Other connection transactions between access requestor 110 and access provider 190 through switch 170 are also possible and can result in different types of connections (e.g., IP, TCP/IP, UDP, and UDP/IP).

Fig. 3 is a block diagram that illustrates logical components of switch 170. As shown, the switch 170 includes the components necessary to detect and prevent a hacker attack on access providers 190. In particular, switch 170 includes a monitoring component 310 and a blocking component 320, which generally include one or more components embedded in software modules within a computing device, but may be embodied in physical devices connected to one another or may be embedded in some combination of software modules and physical devices. In other implementations, the components illustrated in Fig. 3 may be resident on an access provider 190.

The monitoring component 310 is structured and arranged to monitor the computer system 100 for connection transactions between the access requestor 110 and the access provider 190. The blocking component 320 is structured and arranged to deny access by access requestors 110 at one or more particular IP addresses to access providers 190 for a configurable period of time. Thus, attacks launched by any particular IP address may be prevented from reaching access provider 190. In other implementations, the blocking component 320 may be used to deny access by access requestors 110 at one or more

particular IP addresses to access providers 190 for a period of time after a most recent connection transaction has been initiated by the access requestor 110 from that IP address. Thus, the period of time that the access requestor 110 from any particular IP address is denied access does not begin decrementing until after that access requestor 110 from that IP address stops attempting to establish a connection with the access providers 190. To this end, the blocking component 320 may be programmed not to completely deny access by the access requestor 110 at a particular IP address, but instead to allow the monitoring component 310 to continue to monitor the connection transactions initiated between that access requestor 110 at that particular IP address and the access providers 190.

Referring to Fig. 4, the monitoring component 310 may include a detection component 410, a counting component 420, and a comparing component 430. The detection component 410 is structured and arranged to detect at least the initiation of connection transactions between the access requestor 110 and the access provider 190. For example, where the access requestor 110 includes one or more clients and the access provider 190 includes one or more hosts, the detection component 410 is capable of detecting connection transactions between at least one client and at least one host. The detection component 410 is generally programmable and capable of recognizing when a connection transaction is initiated by an access requestor 110. For example, detection component 410 may be programmed to recognize the initiation of a connection transaction used to establish a TCP connection between an access requestor 110 and an access provider 190. In this example, detection component 410 may be programmed to recognize the exchange of one or more messages (e.g., SYN REQ, SYN ACK, ACK) communicated between the access requestor 110 and the access provider 190 as indicative of the initiation or completion of a connection transaction. Additionally or alternatively, detection component 410 may be programmed to recognize connection transactions based on other criteria, or other connection transaction types altogether.

The detection component 410 generally includes an identifying component that is structured and arranged to identify the IP address through the use of a header attached to the IP address. For example, during a first connection transaction, access requestor 110 may be routed to a first access provider 190 such that a connection is established with the first access provider 190. During a subsequent second connection transaction, the same access requestor 110 may be routed to a second access provider 190 that is different from the first access

provider 190, such that a connection is established with the second access provider 190. Thus, the number of connection transactions detected by detection component 410 involving a single access requestor 110 may be cumulative with respect to more than one access provider 190. More particularly, where access requestor 110 includes a client and access provider 190 includes a host, the detection component 410 is capable of detecting connection transactions between the access requestor 110 and multiple access providers 190. The detection component 410 is generally connected to the counting component 420.

The counting component 420 is generally structured and arranged to count the number of connection transactions initiated by the access requestor 110 during a configurable period of time. Counting component 420 counts and records the number of connection transactions that were detected by detection component 410. The configurable period of time may be programmed to be any length of time. The counting component 420 may be programmed, for example, to count all of the connection transactions between the access requestors 110 and the access providers 190. Counting component 420 includes a processor and an internal memory for counting and logging the number of connection transactions.

In one implementation , the counting component 420 counts the number of connection transactions initiated or established between a particular access requestor 110 and a single access provider 190. In another implementation, the detection component 410 detects connection transactions initiated or established between one Internet protocol (IP) address and any access provider 190. The counting component 420 counts the number of connection transaction initiated by that IP address during a set period of time.

Counting component 420 is capable of counting the number of connection transactions initiated from an access requestor 110 at a single IP address, irrespective of the access provider 190 with which the access requestor 110 seeks to establish a connection. For example, during a first connection transaction, the access requestor 110 from a single IP address may initiate a connection with a first access provider 190, while during a subsequent connection transaction, the access requestor 110 from the same IP address may initiate a connection with a different access provider 190. Counting component 420 may be configured to count and log each of these connection transactions in association with the IP address.

The comparing component 430 then compares the total number of connection transactions made by the access requestor 110 from the IP address during a period of time to

a threshold number of connection transactions. If the threshold number is exceeded, then the blocking component 320 denies access by the access requestor 110 from that IP address to access providers 190.

The comparing component 430 is structured and arranged to compare the number of connection transactions that involve the access requestor 110 during a configurable period of time against a threshold number of connection transactions, which is also configurable. In one implementation, when the threshold number of connection transactions between the same access requestor 110 and any access provider 190 is exceeded during the configurable period of time, the blocking component 320 denies the access requestor 110 access to one or more access providers 190.

Referring to Fig. 5, the blocking component 320 includes an access preventer 510, a timer 520, and a reset 530. Access preventer 510 is structured and arranged to deny access by the access requestor 110 to the access providers 190 for a configurable period of time. Access preventer 510 is connected to timer 520.

Timer 520 is structured and arranged to measure the configurable period of time during which access preventer 510 denies access by the access requestor 110. Thus, when the comparing component 430 determines that the configurable threshold number has been exceeded by the access requestor 110, the access preventer 510 denies access by the access requestor 110 and the timer 520 measures the configurable period of time during which access is denied. Timer 520 is connected to reset 530.

Reset 530 may be structured and arranged to reset the configurable period of time measured by timer 520 for which an access requestor 110 is denied access if the monitoring component 310 detects a new connection transaction initiated by such an access requestor 110. For example, if an access requestor 110 that is timed out for exceeding the threshold number of connection transactions initiates a new connection transaction during the time out period, reset 530 will start a new time out period during which the access requestor 110 will continue to be denied access.

Referring to Fig. 6, a process 600 is described for securing an accessible computer system, which process 600 may be performed by the systems described above with respect to Figs. 1-5. For instance, process 600 may be performed by a switch 170, by an access provider 190, or by a combination of the two. The method also may be performed by any other hardware device or software device capable of being programmed to receive, process,

and send instructions in the manner described. The process 600 generally includes monitoring a computer system for connection transactions (step 610) and controlling access to access providers 190 if a threshold number of connection transactions is exceeded over a configurable period of time (step 620). Controlling access may include denying access to the access providers 190 for a configurable period of time. The configurable period of time may be set for any duration. For example, the access requestor 110 at a particular IP address may be denied access for a period of time that does not begin until after the most recent attempt by the access requestor 110 to establish a connection.

Referring to Fig. 7, monitoring the computer system for connection transactions (step 610 of Fig. 6) may include detecting connection transactions (step 710), counting the number of connection transactions (step 720), and comparing the number of connection transactions initiated by an access requestor during a configurable period of time to a configurable threshold number (step 730).

Detecting connection transactions (step 710) may include detecting one or more components of the connection transactions from each of one or more IP addresses. For example, detecting may include recognizing one or more components of the connection transactions used to establish a TCP connection between an access requestor 110 and an access provider 190. In this example, detecting typically includes recognizing the exchange of messages (e.g., SYN REQ, SYN ACK, ACK) exchanged between the access requestor 110 and the access provider 190 during a connection transaction. Additionally or alternatively, detecting may include recognizing other connection transactions or their components.

In another example, during a first connection transaction, an access requestor 110 may be routed to a first access provider 190 such that a connection is established with the first access provider 190, and during a subsequent second connection transaction, the same access requestor 110 may be routed to a second access provider 190 different from the first access provider 190 such that connection is established with the second access provider 190. Thus, the number of connection transactions detected in the detecting step 710 may be between a single access requestor 110 and any of several access providers 190, with the detected transactions being cumulative with respect to more than one access provider 190. For example, where the access requestor 110 includes a client and the access provider 190 includes a host, the detecting step 710 may include detecting connection transactions

between the access requestor 110 and multiple access providers 190. In another example, the detecting step 710 may include detecting connection transactions between at least one client and at least one host.

The detecting step 710 is generally followed by a counting step 720, as shown, which may count and log each connection transaction or transaction involving an access requestor 110. The counting 720 generally includes counting the number of times a connection transaction or transaction component between an IP address and a host has been detected during a configurable period of time. In one implementation, the counting step 720 is capable of counting the number of connection transactions initiated or established between a particular access requestor 110 and a single access provider 190, and the number of connection transactions initiated or established between one Internet protocol (IP) address and several of the access providers 190. In general, the connection transactions initiated by the access requestor 110 are counted during a set period of time.

The comparing step 730 includes comparing a configurable threshold number to the number of connection transactions that have occurred between an access requestor 110 at a particular IP address and one or more access providers 190 during a configurable period of time. If the number of connection transactions or detected and counted transaction components are determined to exceed the configurable threshold number, access to the access providers 190 is denied (step 620 of Fig. 6). Otherwise, access is permitted.

In one implementation, detecting the connection transactions (step 710) includes identifying the IP address through the use of a header attached to the IP address. The use of the header to identify the IP address allows a determination of which particular IP address is being used to attack the access providers 190 through a denial of service attack.

Referring to Fig. 8, controlling access to access providers 190 (step 620 of Fig. 6) generally includes denying access by the access requestor 110 to access providers 190 for a configurable period of time (step 810), monitoring the computer system for connection transactions initiated by a blocked access requestor 110 (step 710 of Fig. 7), and resetting or changing the configurable period of time if a new connection transaction is initiated by a blocked access requestor 110 (step 820), then continuing to deny access by the blocked access requestor 110 to the access provider 190 for the reset or changed period of time (step 810).

A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, advantageous results still could be achieved if steps of the disclosed techniques were performed in a different order and/or if components in the disclosed systems were combined in a different manner and/or replaced or supplemented by other components.

In addition, the systems, methods, and techniques described here may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus embodying these techniques may include appropriate input and output devices, a computer processor, and a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor. A process embodying these techniques may be performed by a programmable processor executing a program of instructions to perform desired functions by operating on input data and generating appropriate output. The techniques may advantageously be implemented in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program may be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language may be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as Erasable Programmable Read-Only Memory (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and Compact Disc Read-Only Memory (CD-ROM disks). Any of the foregoing may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits).

Accordingly, other embodiments are within the scope of the following claims.